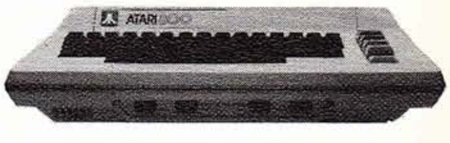
HOOKED ON ATARI COMPUTER KEYBOARD SOCIETY

HACKS



Volume Four, Number Nine **Studium Omnibus Habendum Est** **September1987**

Kitts' Cadre

Kitts is looking for volunteers for the SOUTHERN CALIFORNIA ATARI COMPUTER FAIRE, VERSION 2.0. He wants more than warm bodies. He wants slaves. For more information call him at 818-848-7336.

Such a Deal?

We have discount tickets for sale for the FAIRE. You can give us $3.00 now or $5.00 later. It's your churse.

Broderbund

)ur ST Graphics SIG will be honored with an .xhibition of their fantastic products. Wear your karate shoes. You will also be able to view their Director series of software. Remember Print Shop? Well get ready for the ST version.

Kees' Corps

He has volunteered to man the H.A.C.K.S. booth. So if anybody wanting to do volunteer work and is afraid of Kitts he can talk to kees.

Just in Case

The Dates: 9–19 & 9–20
The times: 10 6 & 10 5

Schedule:

09-01: ST GRAPHICS SIG @ 7:00 PM
09-05: REGULAR MEETING @ 10:30 AM
09-08: ST GENERAL MEETING @ 7:00 PM
09-15: 6502 CLASS @ 7:00 PM
09-22: MIDI SIG @ 7:00 PM

The Tuesday meetings are at Logical Choice.

Hall...Hall the Gang's All Here

Ve now meet at the log cabin at PACIFIC PARK .n Burbank.That is one block North of Victory and three blocks West of Hollywood Way in the 3700 block of Pacific Avenue. Anyone caught wearing a stove pipe hat will be shot. The rest of the stuff stays the same. Doors open at 10:00 and the meeting starts promptly at 10:04 AM.

Programming Techniques

by rootbeers

This column is pretty much dedicated to explaining some of the general data structures and techniques for programming.

This month's exciting episode: programming for speed.

Perhaps you have written a program before which does what you need it to do but it's so slow you might as well take a nap or go to the store while it runs. How do you make it run faster? There are lots of ways to speed up a program, but the best way is to write the program to run fast from the beginning.

The first thing you should really ask yourself is "how important is speed of execution"? My point here is that in general you are either concerned with how fast you can get good results at all (how long it takes you to program and debug it) or you are interested in how fast it will run without regard to how long it takes to program it. Most of the time, you care about both, but think about your speed objective at the earliest point that you can.

The second thing to think about is the language you wish to use. BASIC can be fast if you compile it, but that often means that you should restrict yourself to integer mathematics. Assembler is the fastest to run but takes longest to program. If you know BASIC and assembler consider using an assembly subroutine for the longer tasks.

The third thing to consider is the algorithm you plan to use. A BASIC program with a good technique will beat even an assembler routine with a bad technique. Don't be afraid to do some research, some good algorithms for the more well-known problems are available just by looking in the right college bookstores.

The fourth thing you should consider is the data representation. If you have a choice between a floating point and an integer storage, use the integer when you can. I saw one of my programs run forty percent faster just by a change of data representation.

Now with all those things in mind, you should write the program and get it working, commented, and readable. The first programming criteria is always whether it produces the correct output. Comments and readability are necessary if you want to ask someone else's advice or in case you decide to shelve it for a year while you learn more. Comments and readability are also useful in debugging. Then when it works, and it's already close to as fast as you can write it, since you kept the first four rules in mind when you wrote it, you should make a copy of it and start to reread and hone it for speed.

Given that you kept these things in mind when you wrote your program, when you read through it again to try to make it faster, keep the following things in mind: (1) EVERYTHING takes time. Save any calculations you can use later. Put subroutines in-line whenever possible. Try to imagine what the language is doing for (and to) you (for example, reading an array takes more time than reading a simple variable). (2) "Ninety percent of the execution time is spent in ten percent of the code" is a common addage. Don't bother working on anything but the routine which is slow. (3) Remember that the goal is speed, not size. I have some programs that use half of memory for one table so that the program can go faster. (4) Remember that there is ALWAYS another way to do whatever it is you're doing. Ask others to look over your work and see if they can see a better way.

I hope these ideas help. I have a couple programs that I come back to every other year or so just to see if I can come up with some new way to make it run just a little faster. Probably a silly goal, but it's good sometimes to try to one-up yourself.

The President's Corner

by John King Tarpinian

Well, this is the month. The month of the FAIRE. I cannot wait for it to start. Then again, the minute it is over I will have a nervous breakdown.

As of the writing of this article everything is going more easily than I had hoped for. Exhibitors are buying booths and paying ahead of schedule. All the ACENET clubs are banding together to make sure that this show is a success.

We have discount tickets for you. They are three dollars from the club. The cost at the door is five dollars. Such a deal.

As you know by now the Golden Mall Pavilion is, by now, an intersection. Do not fret we do have a new home. It is the log cabin at Pacific Park. That is one block North of Victory and three blocks West of Hollywood Way in the 3700 block of Pacific Ave.

BRODERBUND will be coming to our ST Graphics SIG this month, September 1st. They will be showing us Print Shop ST, Karateka, Art & Film Director.

My hand is getting sore. So you get a short piece from me this month.

HOOKED ON ATARI COMPUTER KEYBOARD SOCIETY

HACKS

President: John King Tarpinian
Newsletter Editor: Kitts Anderson
Public Domain Editor: Alan Hagge
Telecommunications Editor: Dave Evans
Contributing Editor: Ross Beers
ST SIG Chairman: Tony Lee
Midi-SIG Leader: Glenn Feit
Graphics SIG: Mike Malloy
8 Bit Librarian: Ron Baxter
16 Bit Librarians: Kevin Gallagher and Steven Blackburn
Graphics and Layout: Lew Marchese
Founder: Greg Hiscott
Founded: June, 1981

This newsletter is published by the HOOKED ON ATARI COMPUTER KEYBOARD SOCIETY, HACKS, an informal association of individuals. This group is not affiliated with ATARI or any other commercial organization. Any trademarks or company names are used either with permission or as an informal way of referring to a product or an organization.

Articles are written and donated by members. Opinions expressed in this publication are those of the individual author and do not necessarily represent or reflect the opinions of this group or any other organization or company.

Articles appearing in this newsletter may be reproduced provided credit is given to the author and to the HOOKED ON ATARI COMPUTER KEYBOARD SOCIETY.

Subscriptions to this newsletter are available for $20.00 per year. Issues of this newsletter will be sent at no charge in exchange for any newsletters sent to us at no charge.

Advertising space is available in this newsletter, and can be submitted as camera-ready art or as a pre-printed insert. The advertising will be published in the next available issue upon acceptance of the advertising and payment of the appropriate charge. Advertising rates are: Full Page Ad $40.00, Half Page Ad $25.00, 1/4-Page Ad $15.00, "Business Card" Size Ad $10.00.

Membership dues are $20.00 per year per person. We also have a family rate of $30.00 per family. Payment of dues entitles members to attend general meetings, access to our public domain library, subscription to this newsletter and discounts at participating retailers.

Programming Finesse: Another View

by Alan Hagge

As rootbeers pointed out in his article "Programming Finesse", the cost of software is beginning to far overshadow the cost of hardware, and all signs point to an every-widening gap. A term often heard is the "software life cycle". This refers to the cost of the software over the entire length of time it is used, from initial design, through development and coding, to maintenance and modification (typically the most expensive part).

It is this life-cycle cost that is skyrocketing with no end in sight.

I'd like to take a look at the problem from another angle, one that is just today beginning to catch on. Many of you may have heard of the Ada language. Ada was developed for the military to try to attack the cost problem by facilitating all aspects of the software life cycle. A number of new ideas were proposed to try to make large software projects manageable (by large, I mean over 1 million lines of code!!).

A software design methodology which is quickly gaining favor in industry, and one which Ada supports quite well, is the concept of Object-Oriented Design, or OOD. The basic premise of OOD is to create a compilation unit (in Ada this unit is called a PACKAGE) which contains a particular object and all of the operations which may be performed on that object.

For example, take a stack. A stack has two basic operations, PUSH and POP. In this case, the STACK package would contain the object STACK, and the operations PUSH and POP. A trigonometry package might contain an object called an ANGLE, with operations of SIN, COS, TAN, ARCSIN, ARCCOS, and ARCTAN permissible on the object ANGLE.

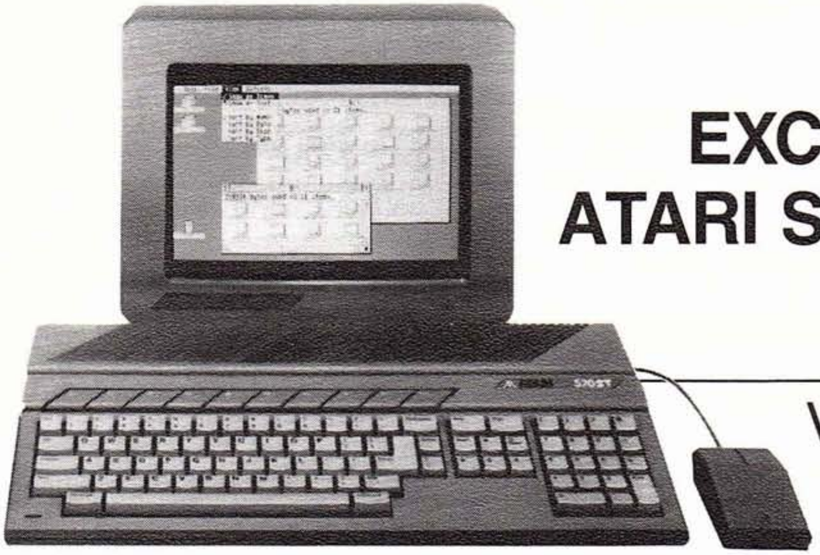
Two key advantages that come from this design approach are low coupling and high cohesion. Coupling is the degree to which two routines depend on each other or on common data shared between both. Cohesion is the degree to which a routine can stand on its own, with no outside programs or data except input and output parameters. Low coupling and high cohesion make for "stronger" (read more useful and easily maintainable) routines.

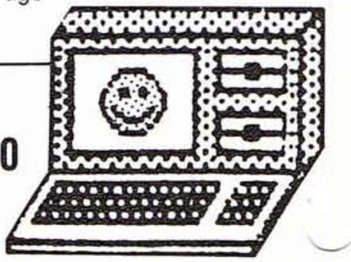
The concept of a package is not necessarily limited to Ada however. You can take advantage of this facet of OOD in your programs as well. If you write in Pascal or C, you can create subroutine files which can either be included or better yet, linked into your main code. The guidelines for these packages are:

1) Put only one type, or a group of related types into a package. If a new type (i.e., enumeration or record) cannot be used, document which type of variable (i.e., INTEGER, REAL, etc.) is to be used with this package.
2) Think of all possible operations you might want to do on those types, and implement them as subroutines and functions within the package.
3 If at all possible, DO NOT use any external variables except those which are passed as parameters.
4) Try to minimize the use of routines not in the package. If you MUST use routines in another package, document that fact at the top of the package, since any changes to the other package might well affect this one, too.
5) By all means, use meaningful variable and function names! X1, Y2 and P are worthless names when you have to look at year-old code.

There are other aspects of Ada and OOD which I think you will find helpful in your coding practices, whichever language you choose. Next time, I'll take about the concept of reusability and the Ada generic program unit.

YOUR ARTICLE BELONGS HERE.





Something to Do

by rootbeers

This column is dedicated to presenting some of the 'classic' and/or bizarre programs and problems of computing in the hopes of encouraging others to try their hand at programming.

One well-known computing diversion is called "Life". Take a piece of graph paper and fill in a few squares. This is called a "Generation" and since you just created it out of thin air, let's call it generation 0. Each filled in box is a "Marker" and the number of markers is the "Population". Each marker can have as many as eight "Neighbors", which are the squares above, below, left, right and diagonal. Now take a second sheet of graph paper and fill it in accordance to the following rules: if the corresponding box on the first sheet of paper has exactly three neighbors, then fill it in. If the corresponding box on the first sheet has two neighbors then fill it in only if it was filled in on the first sheet. Do this for all squares of the graph paper, and you have created generation 1. See generations 0 through 4 for examples. Pretty tedious, huh? Well, that's where the computer comes in. The goal is three-fold: do the necessary calculations, do as large an area as possible, and do it as fast as you can.

Now someone is going to ask "why do it at all?" and the answer is really the name of the column, but to be honest, the patterns which emerge from the simple rules are truly incredible. For instance, there are "gliders" which move across the graph paper. There is even a "glider gun" which sends off a new glider every sixty-some-odd generations.

I have not yet seen what happens when one uses similar but different rules or even different shapes. The possibilities are enormous and it is a good application for a graphics-oriented machine. One thing I have become interested in is the possibility of creating a set of rules for a three dimensional life.

Anyway, ask around and I'm sure you'll find someone's home-brewed version of life to play with. If you can't though, try your hand at writing it yourself. You've got nothing to lose but your piece of mind, your self-respect, and a few night's sleep. When you've lost all that, you'll fit right in with the other hackers.

ACENET

ATARI
COMPUTER
ENTHUSIASTS
NETWORK

CHARTER MEMBER

ST MIDI SIG

by Ben Di Tosti

The 3rd H.A.C.K.S. ST-MIDI SIG meeting which occurred the last Tuesday of June had been expecting to see a demo of the STEINBERG Pro-24 sequencer. Due to the fact that the National Assoc. of Music Merchants (NAMM) show was scheduled that weekend, the people who represent STEINBERG decided to leave earlier for Chicago and therefore cancelled their appearance. By the time you read this, they should have done their presentation at the August meeting. The meeting was certainly not a washout because we again had the opportunity to benefit from the wealth of experience that meeting moderator Glen Feit brings to these MIDI gatherings.

The 4th ST-MIDI SIG meeting took place the last Tuesday of July, at LOGICAL CHOICE FOR COMPUTING. A brief recap on the above NAMM show occurred, and it seems that a complete floor was dedicated to music software. The approximate count of computers was as follows: IBM—21; MAC—20; Atari ST—55; Others—5. Atari made headlines as the first personal computer company ever to sponsor an exhibit at NAMM. The highlight of the show, it seems, was the introduction of the ALESIS drum machine, HR 16. Features include 48 various drum sounds of the past 50 years that can be stored as 100 patterns and 100 songs. Every sound is tuneable, touch sensitive, and assignable to any pad. MIDI is implimented. Are you ready for the price? $449! But you gotta wait til the late fall (early next year?) to purchase it.

Other companies displaying their MIDI-softwares were AEGIS, AKAI, BEAM TEAM, COMPU-MATES, DIGIDESIGN, DR. T, ELECTRONIC ARTS, HYBRID ARTS, MICHTRON, PASSPORT, STEINBERG, and TWISTER. For an in depth report of the above consult the August issue of ST Applications.

After the rap on NAMM, we were given a rather lengthy demo on some of the new updates which will be released this fall by HYBRID ARTS, in particular, SMPTE track. Programmer Stephan Daystrom, who incidentally has attended all of our MIDI meetings, proceeded to describe those features he has been crafting of late, of which the major one is the graphic editor. This consists of a split screen showing a selectable range of notes—in time—on an horizontal axis, together with a four octave vertical display. The upper right screen area consists of individual highlighted information correlating to whichever note the mouse is pointing to on the grid, such as note-name, start & stop time, attack and release velocity etc. Clicking on a note selects it for editing in which the aforementioned values may be altered.

Additionally, you may grab a whole range of notes with the mouse and then zoom in and play, transpose, or delete that particular section. Stephan claims that the user will also be able to "paint" a range dynamically to affect crescendos and diminuendos, as well as redraw other controller effects such as pitch-bend, after-touch and MIDI volume.

Other additions are a fast forward/rewind screen button similar to that of tape machines, and localized track editing features which include quantizing, transposition, and velocity editing.

Also planned for the fall release is a rather useful feature that will allow the user to establish and change time signatures throughout the track layout. Stephan claims that measures, beats and ticks will now be displayed.

Shadow Track allows you to copy a track to another track in order to permit it to be set to a different MIDI channel (thus a different layered patch-sound) or a different track delay. Another advantage is that the user can assign a looping structure to two locations. And all this is done on a temporary basis so that there is no need for additional memory.

Another new feature will allow the user to record system-exclusive messages within the sequence. This would allow for shorlt messages to aid in future MIDI spec extensions such as would allow for larger patch-bank changes.

Next, Stephan elaborated on GenPatch ST, which is a powerful stand alone program that can handle a complete MIDI patch library and works with any MIDI device: synths, drum machines, effect units, and samplers. It has a configuration library which is constantly expanding and is available for download off the HYBRID ARTS BBS. It also has a built-in configuration editor to allow the user to form new configuration files for any MIDI instrument, including future products. The real-time MIDI data display screen is quite instructive in showing the actual code that occurs when you play a note or send a patch dump.

Finally, there is the GenPatch Desk Accessory, which is a part of SMPTE-Track and SYNC-Track and loads various synth patches from within the sequencer. Therefore, when you load a new song the GenPatch File Auto-injector will also load the proper configured patch for that song and designate which synthesizer modules will be active.

One comment on the MIDI meeting. It seems that there is a good mixture of apprentices and seasoned MIDI users. However, I can't help but notice the continued perplexed looks of some who probably feel they are in over their head. I can only say that MIDI is BIG and growing all the time. Please learn to be a part of it. Do your homework by reading the few good books on MIDI as well as the many magazine articles. Do not be afraid to ask questions. Also remember, you learn by doing! Do buy an inexpensive synth and moderately priced ST sequencer, if you haven't done so yet. One thing will lead to another and before you know it, you'll be a part of this wonderfull new technology.

Ramdisks Continued

by Kees Jongsma

A ramdisk is a portion of ram memory set aside to be used as a disk storage area. It provides a super fast access time and easy file manipulation. In the June issue of Hacks I covered the Ramdisk programs available in our HACKS PUBLIC DOMAIN LIBRARY and described how to use them. In preparing that article I put together a disk called Ramdisk Organizer that essentially provided for auto setup of ramdisks and documentation for their use. This disk may not yet be in the HACKS PUBLIC DOMAIN LIBRARY and if not I will be happy to provide it to those interested. If you haven't tried working with a ramdisk I highly recommend that you do as they are very useful.

One of the questions that I received concerning ramdisks is "Why so many different kinds?". Each of the ramdisks have slightly different features. These features range from how they boot to size limitations and resetability. Some of the ramdisks will not work with certain software (watch out degas users) and it is recommended that you try out a piece of software before using it for a serious application.

I recently found a great application for ramdisks and MAGIC SAC. Since I have a single drive ST with a MEG of memory, booting up a large Macintosh application required the swapping of the systems disk and the applications disk a number of times. This is a real drawback for serious users (such as myself) and I knew that if a ramdisk could be made to look like the boot disk that my problem would be solved. I hooked up to Genie and found that my problem had already been solved with the use of a program called RAMSTART.

By inserting this program into the system folder it can be made to auto boot and dump the rest of the system folder into the ramdisk. I am using Ramstart version 1.2 and version 1.3 should also work. I recommend the following procedure to set up your MAGIC SAC ramstart disk.

1. Prepare a new magic sac formatted disk
2. Copy system folder onto the disk including ramstart in the system folder
3. Open the system folder and click on Ramstart once
4. Go to the menu bar and select "set startup" from the specials menu
5. After the disk is done spinning you are done.

At this point you reboot the system and use your new Ramstart disk as the startup Disk. As mentioned above this ramstart may not work with all application (although I haven't found any that do not work yet) so be careful before any "real work".

Norm's Super Duper All-in-One Floppy System

Are you ready to add another floppy drive to your ST? Handy with tools? If you are adventurous you might want to try and save some money by building your own floppy system.

Recently I decided to upgrade from single-sided drives to double-sided for my 520ST. Since I don't have a built-in drive, I figured that I could build two, external double-sided drives into one case. I purchased two 3.5" drive mechanisms and a standard case with power supply designed for a fulll height 5.25" floppy drive. These cases are cheap and widely available.

Fortunately, two 3.5" drives stacked, one on top of the other, are exactly the height of one 5.25" drive. So they fit nicely into the case. The power supply that is built into the case provides 5 volts and 12 volts, both at better than 1 amp. This is enough power for two and maybe three 3.5" drives. After they are mounted in the case, there is about one inch on either side of the drives that is open and unused. I just filled this with something decorative.

The steps of this project are to physically mount the drives in the case, to connect the power lines and the signal lines and finally, to make a cable to connect the drives to the ST. The advantage of packaging your floppies this way is that you ony have one box with one AC line cable and one signal cable going to the computer.

MOUNTING

Mounting the drives in the box takes a little thought. The drives are rectangular boxes, the connectors facing one end and the front bezel on the opposite end. The bezel is the painted part with the slot to insert disks. It should protrude beyond the front of the case. Or "they" should, since the drives are one above the other. I placed one drive on the floor on the case, with the bezel hanging over the front. There is an inch left over to either side and several inches behind.

You could drill holes in the floor of the case to fasten the drive down, but I found it better to fashion some L-shaped brackets that fasten to the floor, in that spare one inch to the sides. These brackets are tall enough to reach about a half-inch above the bottom drive. The height really depends upon where the mounting holes are placed on the drives you obtain.

There are several threaded mounting holes on the sides on the drives. I drilled holes in the brackets in line with these holes. Then, at a measured distance above these holes, I drilled a second set. Now the second drive can be lined up with these upper holes and zip! They're mounted.

The mounting holes require METRIC screws (3 mm)! You can force 4-40's into the threads, but that's crude. The electronic parts stores I frequent do have little sacks of 3mm hardware (from General Cement) which contain many more than you need.

Signals

It is far easier to find 15-pin "D" connectors than the DIN 14-pin connectors that Atari uses for the disk drives. Therefore, I mounted a 15-pin "D" female to the back panel of the case. I then took a one foot piece of 34 conductor ribbon cable and mounted two 34-pin insulation displacement connectors, one at an end and the other about 2 inches from the first. They are going to plug onto the drives, so place them so you can get them both connected comfortably.

At the other end of the ribbon cable, the individual wires are separated. The ones that will be connected are stripped and tinned, the others are cut off. See Table I.

More Signals

Looking at the printed circuit board of the floppy drive, in the vicinity of the 34-pin connector you will find two rows of vertical pins with a little, plastic-bodied rectangle plugged across one pair. Letting on the board will indentify the pins as "DSO, DS1", etc. Usually this shorting-plug is connecting the DSO pins (Drive Select 0). In this position, the drive will respond as drive "A". Pull off the plug on one drive and plug it onto the pins marked "DS1". This drive will respond as drive "B". Which drive is "A" and which is "B" is your choice.

Now look at the components near the 34-pin connector carefully. One DIP (dual in-line package) or SIP (single in-line) is not soldered to the board, but is socketed. The printing on the board near it may say "Terminators". These are the pull-up resistors of the signal lines. They serve a very specific purpose and must be connected only at the connector which is at the end of the line. That's the reason they're socketed. When the ribbon cable is in position, one drive is at the end. Leave the resistor pack in that drive, but remove it from the other drive.

TABLE I

Ribbon Wire	15-Pin Socket	Signal Name
8	4	Index Pulse
10	5	Drive 0 Select (Drive A)
12	6	Drive 1 Select (Drive B)
16	8	Motor On
18	9	Direction In
20	10	Step
22	11	Write Data
24	12	Write Gate
26	13	Track 00
28	14	Write Protect
30	1	Read Data
32	2	Side Select
Odd*	3,7	Grounds
	15	Net used

* All odd numbered wires of the ribbon cable are connected to ground. Choose two convenient ones to wire to pins 3 and 7.

The cable from the ST to this system is made from a standard Atari floppy cable or an equivalent extension cable. Cut off one of the fourteen pin DIN connectors and mount a 15-pin connector at that end. Except for pin 15, which is not used, each pin is connected one-to-one to the same numbered pin.

Power

There is a three wire cable from the power supply. One wire is ground (almost certainly, the black one), one is +5 volts and the other +12 volts. Use a meter to find out which. These terminate in a 3-pin Molex connector. Cut it off and discard. The power for the floppy drives connects through a four pin connection on 0.1 inch centers. I could not find the exact matching piece in the electronics stores, so I got some similar ones. They had ridges on one side for polarization which prevented them from pushing on the drive. Five seconds with a knife whittled them off and they were no longer a problem.

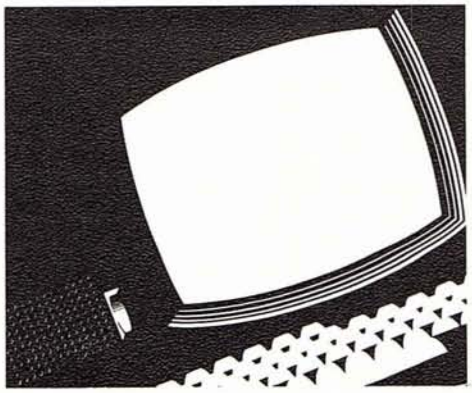
The power connection at the drive consists of four wires. The center two are ground; one outside pin is 5 volts, the other is 12 volts. Looking at the pins with the drive connector at the top, the pin at the right end is 5 volts. The sure way is to use an ohm-meter and check for continuity to the power pin on one of the IC's. That'll tell you which is +5V, the other end is +12V.

You have to make up two power connectors, one for each drive. Tie the grounds (four wires) together and splice to the ground wire from the power supply. Similarly, tie the two +5V wires and splice to the power supply; and again with the +12V wires.

Done

That's it. You're ready to fire it up! This project has been tried and it works. Go to it!

Norm Weinress



Cut and Paste

y Steve Blackburn

Newspapers, magazines, and just about everyone else has used the cut and paste method of page layout to create the different types of junk mail that we all receive daily. They take pictures, text, and graphics that they have created from various sources and physically paste them up on the pages that will eventually sent to the printers. Many if not most still do it this way. Computer generated pages are not yet common day occurences.

With programs like printmaster, Degas Elite, and others combined with your favorite word processor, it is very easy for you to do the same thing.

Starting with a blank page, you design it with the tools you have available. Degas and Printmaster are great for making letterheads and logos. These programs are also well suited for making graphics that can be placed within your text. By a simple process of cutting out the images and pasting them to the blank pages, along with text, you can create an almost endless variety of finished products that can be used for your own personal use or created for small companies for a profit. Some word processors do not have the ability to create double columns, but by printing out your documents at 20-30 words wide and then asting them up, you can achieve the same end esult. Graphics and lines from drawing rograms can be added as well. One way to achieve a higher quality graphic image is to create it on a large scale (full page) and then taking it to your local copy shop and having it reduced. You can even improve the image by first taking a lack marking pen and darkening the areas you think might not reproduce well, and as we all know, printers are notorious for leaving white lines in the printout. Once all your corrections and improvements are made, have it reduced to the size needed and then paste it up with the text you have created. Once all of this is on one piece of paper, take some white-out material and go over the areas of the paper that might cause problems (the edges of the individual pasteups are your biggest problems). Now take it down to the local print shop and have them make you one copy. Use whiteout for any other areas that are causing you problems, and then have them run off as many copies as needed.

I have had a lot of fun doing things this way, and have created pages that would compare to many of the commercially available methods of creating flyers, ads, etc. Your imagination and design skills are all that are keeping you from creating truly professional finished products.

I enjoy graphics and creating printed copy, and anyone else who is interested in the same is 'elcomed to leave me messages on the ST UNGLE BBS (213) 254-9534 (leave messages ıor Steve Blackburn or The Bear on the graphics/program help base).

Hope to be hearing from you...

PD Spotlight

by Alan Hagge

I'm back! Now that I'm lost in the desert, I'll probably only be able to get out an article every other month or so. Bear with me, and if you can't stand it, write one yourself!

This month I'd like to tell you about a home loan calculation program called Home Loan Analysis. It was written by J. F. Skinner for Antic Magazine and available on the Finance 1 disk in the library. This program allows you to do "what if?" calculations on a home loan, such as "how much less will my payments be if I put 20% down rather than $5000.00?" Or "how much interest can I save by taking out a 15 year loan rather than a 30 year loan?"

An easy-to-use main menu prompts you for such variables as the purchase price of the house, the down payment, the interest rate, and the length of the loan. The program has enough smarts to calculate some variables based on those you give it. You can then easily change any number and have the computer calculate the changes. As an added feature, the program can compute the future value of your property, based on the buying price and an appreciation rate.

The most important part of the program allows you to see an amortization of your loan. You can look at the whole thing or any year within. It also shows the cumulative totals of principal and interest for each year. All of the information can also be sent to your printer.

Granted, there are other public-domain loan programs out there, and most do pretty much the same thing, but I found the Home Loan Analysis program to be the easiest to use. My sole gripe is that it will sometimes recalculate the wrong variables in the loan formula ("I wanted my payments to do down, not the purchase price to go up!!!"). But all in all a useful program for anyone looking into real estate.

The Mouse & the Handicapped

by John King Tarpinian

Now that I have lost the use of an arm, for the next six months, I have learned to appreciate my mouse & even GEM. Just try Control/ Alternate/Delete on an IBM using one hand. You need to put a pencil between your teeth to do it. A single click of the mouse button can usually take care of most operations.

Even with a word processor, the GEM and the pull down menus make typing relatively easy. I am a touch typist. I have had to learn to type all over again. I use the eraser end of a pencil as a typing helper. The pencil allows me to reach across the keyboard more quickly.

Of course using a joystick is a problem. But, games such as BARBARIAN are fully mouse controlled with no keyboard input needed either. Forget about Flight Simulator II. Of course I cannot play that with three hands.

This is just something I have become aware of that I never thought of before. We take so many things for granted. Those of us that have the use of all our limbs should be grateful and more considerate of those that do not.

Speaking Like a Pro

by rootbeers

In this column I attempt to present some of the actual jargon I've actually heard used by actual computer programmers. Generally a bit on the silly side, but after all, fun is why we're here.

Pel—This is IBM-ese (...IBM-ese: kind of sounds like a laxative, doesn't it?...) for pixel, which is the smallest picture unit which can be addressed or altered.

Set—a coallection of things which has no order associated with it but has the property of "uniqueness". That is, an item is either in the set or not in the set but there are no duplicates.

Bag—a bag is the same as a set except that it may have duplicates.

Sequence—a sequence is a collection of things which have an associated order.

Confused—a euphamism for "wrong" usually used in referring to inaccurate statements made by one's superiors.

Sleeper Hold—The mechanism by which one program during the course of its failing causes "normal" programs or files or hardware to fail.

Vent one's spleen—a rather graphic phrase connoting getting something off your chest. Note the rather Picasso-istic change of anatomy.

Line of Death—The highest address of usable memory. Sometimes used to mean simply the end of available memory.

Oxymoron—The guy who "does" fries at a fast food place.

HACKS

HACKS
HOOKED ON ATARI COMPUTER
KEYBOARD SOCIETY

6055 Cahuenga Boulevard, #2
North Hollywood, CA 91606
818-760-1831

S.L.C.C.
NEWSLETTER EXCHANGE
P.O. BOX 1506
SAN LEANDRO, CA 94577-0374